Conditional Notification

Some Examples

Michael Koster

13 December 2017

Conditional Notification

- Controls for Asynchronous Notification
- Results in a sequence of notifications that conforms to a filter specification
- Consist of time and value threshold controls
- Examples from IETF CoRE, OMA LWM2M, and OCF

IETF – CoRE Dynlink (draft)

- Split from the CoRE Interfaces Draft
- OMA LWM2M follows the vocabulary
- Time threshold controls are "pmin" and "pmax"
- Value threshold controls "gt", "lt", and "st" apply to scalar values only; e.g. numbers
- Band mode for notification control based on value range

Threshold controls

- **pmin** is the minimum time interval between notifications, even if data are changing more frequently
- **pmax** is the maximum time interval between notifications, even if data have not changed
- **st** (step) is the minimum reportable change in data value, relative to the most recent notification
- **lt** (less than) is the low value reporting threshold
- **gt** (greater than) is the high value reporting threshold

Reporting mode

- Simple threshold mode (default)
  - report whenever lt or gt thresholds are crossed in either direction
- Band mode (when the "band" parameter is included or set to true)
  - report when the sample is between lt and gt values, according to pmin, pmax, and st
  - gt may be less than lt, resulting in reporting only outside the band
  - some systems may allow more than one band to be active at a time

Conditional Observe

- Conditional Notification controls may be configured in URI parameters of Observe operations

```
GET coap://[fdfd::13]/remote?pmin=1&pmax=300&st=1
```

Minimum reporting period

Maximum reporting period

Reportable value change (step)

CoRE Dynlink Bindings

- Dynamic Link Bindings are hyperlinks which configure dynamic data transfers
- Links with the "boundto" link relation point to the source of notification data
- Notification data are transferred to the link context
- Notification parameters are encoded as link target attributes
- Link Bindings are created and stored in one or more binding tables (collections) at the server with rt=core.bnd

CoRE Dynlink example (local update)

```
{
  "anchor": "/local/resource",
  "rel": "boundto",
  "href": "coap://[fdfd::13]/remote",
  "pmin": 1,
  "pmax": 300,
  "st": 1
}
```

Resource to be updated

Resource to be observed

Minimum reporting period

Maximum reporting period

Reportable value change

CoRE Dynlink example (remote update)

```
{
  "anchor": "coap://[fdfd::13]/remote",
  "rel": "boundto",
  "href":"/local/resource",
  "pmin": 1,
  "pmax": 300,
  "st": 1
}
```

Resource to be updated

Resource to be observed

Minimum reporting period

Maximum reporting period

Reportable value change

OMA LWM2M Notification Control

- OMA LWM2M Notifications are sent from the LWM2M Device to some application handler in response to CoAP Observe
- LWM2MConditional Notification parameters are configured on an object or resource using the "Write Parameters" operation:

```
PUT coap://[fdfd::13]/3303/0?pmin=1&pmax=300&st=1

(empty payload)
```

OCF Conditional Notification

- OCF Conditional Notification is a resource type that is composed with another resource in order to control notification (one setting per resource instance)
- Properties are minnotificationperiod, maxnotificationperiod, and threshold
- The definitions correspond to pmin, pmax, and step, respectively
- Threshold uses the same units and scale as the measurement data

OCF Conditional Notification

```
POST /example?rt=oic.if.notification.conditional
(payload)
{
  "minnotificationperiod": 1,
  "maxnotificationperiod": 300,
  "threshold": 1
}
```

Minimum reporting period

Maximum reporting period

Reportable value change

Notification Patterns

- Observe with a sequence of responses, e.g. CoAP, eventsource, websockets
- Publish/Subscribe using a broker, e.g. MQTT
- Web Hook, e.g. HTTP PUT/POST to URI, external update link binding
- Notification queue
- With or without created subscription resources

Subscription creation

- Sometimes a resource already exists from which notifications can be obtained using observe (or retrieve)
- Sometimes a resource is created, from which notifications are them obtained through some address or handle – this resource may be deleted when the workflow terminates
- The websocket notification design uses this pattern in order to share a websocket connection over different notifications

Subscription Creation

- With Pub/Sub, a new topic may be created on which to receive the notifications
- A Dynamic Link or Web Hook is a sort of created subscription that pushes updates to another server
- A "monitor" resource may be created and updated using a dynamic link, from which to obtain notifications using Observe
- Such a monitor resource may also store notifications to be retrieved in bulk

What is an Event?

- An event is a state change that conveys more than a simple property state change
- Separate the semantics of an event from the mechanism for receiving Event notifications
- For example, a button may emit shortpress, logpress, and doublepress events that are related to change patterns in the button state over time
- Events and Property change notifications can use the same mechanisms – observe, pubsub, web hooks, with and without created subscriptions